An Example Routine Calling the MATLAB Implementation of the Anderson-Moore(AIM) Algorithm

Gary S. Anderson and Rebecca Zarutskie
Board of Governors
Federal Reserve System
Washington, DC 20551
Voice: 202 452 2687
Fax: 202 728 5892
ganderson@frb.gov

August 19, 1998

Abstract

This paper describes how to use the MATLAB implementation of the Anderson-Moore Algorithm[1, 2, 3] for imposing the saddle point property in dynamic models. The paper uses a simple two-equation firm value model to demonstrate model construction and solution.

Contents

1 Introduction and Summary

This paper describes how to use the MATLAB implementation (currently downloadable at http://federalreserve.gov/pubs/oss/oss4/ code.html) of the Anderson-Moore Algorithm[1, 2, 3] for imposing the saddle point property in dynamic models. The paper uses a simple two-equation model to demonstrate model construction and solution.

2 The Firm Value Model

This paper uses AIM to investigate the solution of a simple linear model, first presented in [3], which describes the value of a firm.

The model consists of two equations:

$$\begin{aligned} V_{t+1} &= (1+r)V_t - D_{t+1} \\ D_t &= (1-\delta)D_{t-1} \end{aligned} \tag{1}$$

where V is the value of the firm, D is the dividend, r is the interest rate, δ is the growth rate of the dividend (here, negative).

3 Model Representation and Preprocessing

Describe the linear model using the MDLEZ syntax and save the model in a file, here called firmvalue.mdl

MODEL> Provides a name for the model. This does not affect the AIM calculations.

ENDOG> Provides names of the endogenous variables. In the AIM formulation, modellers must completely describe the long run behavior of the system. As a result, all variables are endogenous. “Exogenous” variables must have, at least, a simple forecasting equation.

EQUATION> Provides a name for the equation. This does not affect the AIM calculations.

EQTYPE> Specifies the type of equation. This does not affect the AIM calculations. The keyword STOCH indicates the equation has a stochastic error term. The keyword IMPOSED indicates the equation has no error term.

EQ> Provides the model equation.

"firmvalue.mdl" 3 ≡

```
MODEL> FIRMVALUE

ENDOG>
                    V
                    DIV


EQUATION> VALUE
EQTYPE> IMPOSED
EQ>     LEAD(V,1) = (1+R)*V - LEAD(DIV,1)

EQUATION> DIVIDEND
EQTYPE> IMPOSED
EQ>     DIV = (1-DELTA)*LAG(DIV,1)


END
```

◊

Depending on whether you are working in UNIX or Windows, download the appropriate MATLAB preprocessor. Run the model file through the model preprocessor. If working in UNIX you would type:

```
mdlezAimMatlab.exe firmvalue.mdl
```

This creates two matlab functions, compute_aim_data.m and compute_aim_matrices.m.

4 Using the Anderson Moore Algorithm

Create a program to call AIM and the supporting routines in compute_aim_data.m and compute_aim_matrices.m The small numbers to the right of the descriptions refer to sections in the paper describing the MATLAB code.

To the path add the directory in which the downloaded MATLAB code and the two functions from running the preprocessor have been saved.

```
>>path(path,'/irm/home/you/matlab')
```

where /irm/home/you/matlab is an example directory in UNIX.

"firmvalue.m" 4 ≡

```
[param_,np,modname,neq,nlag,nlead,eqname_,eqtype_,endog_,delay_,vtype_] = ...
        compute_aim_data

%Choose values for the parameters R and DELTA
R = 0.10;
DELTA = 0.60;

compute_aim_matrices

% Construct H matrix from cofg, cofh

[rh,ch] = size(cofh);
[rg,cg] = size(cofg);
hmat = zeros(rh,ch);
hmat(1:rg,1:cg) = cofg;
hmat = hmat + cofh;

%Start calling AIM routines
[zb,hb,zf,hf] = numericBiDirectionalAR(hmat)
tm = numericTransitionMatrix(hf)

%Computing the Q matrix
theq = numericAsymptoticConstraint(zf,zb,hf,1)

%Computing the B matrix
theb = numericAsymptoticAR(theq)
```
◊

This code produces the following output:

```
param_ =

R
DELTA


np =

     2


modname =

FIRMVALUE
```

```
neq =

     2


nlag =

     1


nlead =

     1


eqname_ =

VALUE
DIVIDEND


eqtype_ =

     1
     1


endog_ =

V
D


delay_ =

     0
     0


vtype_ =

     0
     0

hmat =
```

```
         0         0   -1.1000         0    1.0000    1.0000
         0   -0.4000         0    1.0000         0         0

zb =

   -1.1000         0    1.0000    1.0000


hb =

   -1.1000         0    1.0000    1.0000         0         0
         0   -0.4000         0    1.0000         0         0


zf =

         0   -0.4000         0    1.0000


hf =

         0         0    0.7778    0.2828   -0.7071   -1.4142
         0         0    0.7778   -0.2828   -0.7071    0.0000

tm =

   (1,3)       1.0000
   (3,3)       1.1000
   (2,4)       1.0000
   (3,4)      -0.4000
   (4,4)       0.4000

theq =

         0   -0.4000         0    1.0000
    0.0000   -0.0000    1.2095   -0.6911

theb =

   -0.0000    0.2286
         0    0.4000
```

.

A Files

"firmvalue.m" Defined by scrap 4.
"firmvalue.mdl" Defined by scrap 3.

B Macros

C Identifiers

References

[1] Gary Anderson. A reliable and computationally efficient algorithm for imposing the saddle point property in dynamic models. Unpublished Manuscript, Board of Governors of the Federal Reserve System. Downloadable copies of this and other related papers at http://irmum1.frb.gov/˜m1gsa00/summariesAbstracts.html, 1997.

[2] Gary Anderson and George Moore. An efficient procedure for solving linear perfect foresight models. Unpublished Manuscript, Board of Governors of the Federal Reserve System. Downloadable copies of this and other related papers at http://irmum1.frb.gov/˜m1gsa00/summariesAbstracts.html, 1983.

[3] Gary Anderson and George Moore. A linear algebraic procedure for solving linear perfect foresight models. Economics Letters, 17, 1985.

[4] Michael W. Berry. Large scale sparse singular value computations. University of Tennessee, Department of Computer Science, 1996.

[5] Olivier Jean Blanchard and C. Kahn. The solution of linear difference models under rational expectations. Econometrica, 48, 1980.

[6] Gene H. Golub and Charles F. van Loan. Matrix Computations. Johns Hopkins, 1989.

[7] E. V. Krishnamurthy. Parallel Processing: Principles and Practice. Addison-Wesley, 1989.

[8] David G. Luenberger. Time-invariant descriptor systems. Automatica, 14:473–480, 1978.

[9] Ben Noble. Applied Linear Algebra. Prentice-Hall, Inc., 1969.

[10] J. Taylor. Conditions for unique solutions in stochastic macroeconomic models with rational expectations. Econometrica, 45:1377–1385, —SEP—77.

[11] C. H. Whiteman. Linear Rational Expectations Models: A User's Guide. University of Minnesota, 1983.